<u>REMARKS/ARGUMENTS</u>

Claims 1-22 are rejected under 35 USC 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5

Claim 1:

(a) It is unclear what "the RAID is allowed to access" actually means.

(b) It is also unclear what "before completion of the consistency initialization is completed ... the consistency initialization is allowed to start."

10   The applicant has amended claim 1 to state, "...before the consistency initialization is completed, the RAID is allowed to be accessed while the consistency initialization is in progress". In this way, the claim wording now matches the interpretation as suggested by the Examiner. No new matter is entered.

15

Claim 10 is dependent upon claim 7 but there is no antecedent basis for certain elements mentioned in claim 10.

The applicant has amended claim 10 to prevent the problems identified by the Examiner in addition to the reasons explained below. No new matter is

20   entered.

Claims 14, 15, and 16 recite "the regional initialization" but there is no antecedent basis for this term in these claims.

The applicant has amended claim 1 to introduce the phrase "a regional

25   initialization", thereby providing antecedent basis for the phrase in the later claims. No new matter is entered.

Claims 1, 3-16, and 19-21 are rejected under 35 USC 103a as being unpatentable

30   **over DeKoning et al. (Pat 6,467,023);**

9

Claims 2 and 17-18 are rejected under 35 USC 103a as being unpatentable over DeKoning et al. (Pat 6,467,023) as applied to claim 1 above, and further in view of Humlicek et al (Pat 5,822,782); and

5

Claim 22 is rejected under 35 USC 103a as being unpatentable over DeKoning et al. (Pat 6,467,023) in view of Humlicek et al (Pat 5,822,782) as applied to claim 2 above, and further in view of Rezaul Islam et al (Pat 6,282,670).

10   In order to make the claims of the present invention more patentably distinct from the cited reference, in addition to the amendments identified in the above remarks, the applicant has amended the claims of the present invention as shown in the above claim listing. No new matter is entered.

Firstly, the applicant wants to explicitly explain the invention of Dekoning 15 (US6467023). According to abstract of Dekoning, Dekoning relates to methods and associated structure for enabling immediate availability of a disk array storage device. In particular, the methods and associated structure of the present invention permit access to a logical unit of a storage system immediately following creation of the logical unit. Initialization of the logical unit to initialize redundancy information 20 therein proceeds in parallel with host system access to the storage space of the logical unit. The initialization process maintains **a boundary parameter value indicative of the progress of the initialization process.** Storage space above the boundary has had its redundancy information initialized while storage space below the boundary has not. Where an I/O request is entirely above the boundary, it is processed normally in 25 accordance with the management of the logical unit. **Where part of an I/O request is below the boundary, it is processed in a special manner that assures integrity of the redundancy data.** In addition, "initialized" refers to consistency of stored data and related redundancy information, according to column 6, lines 65-67.

In view of invention of Dekoning, the key points of Dekoning can be 30 summarized as follows:

(1)"boundary concept"- **Initialization boundary parameter value** 124 is used to indicate the degree of progress in the initialization of progress in the initialization of a newly created LUN, which means that **the boundary parameter value is used to distinguish portions of a LUN that have been initialized from those portions yet to be initialized.**

According to Dekoning, the LUN is divided into a plurality of portions, and the portions of the LUN are initialized sequentially one after another from top to bottom. Therefore, the initialization boundary is recorded by using a local variable CURRENT and the state information thereof is saved as checkpoint data. Before the completion of the initialization, if the initialization is interrupted due to power loss or something else, the initialization process can be resumed after the interruption cause disappears, by using the checkpoint data. [See column 7, lines 42-46: "element 300 is first operable to set a local variable CURRENT to point to the first physical portion of storage space in the LUN (i.e., to the start of the LUN);" and "Elements 302 through 310 are then iteratively operable to initialize each subsequent portion of the LUN." Also see column 8, lines 1-6: "Element 307 then saves state information regarding the progress of the LUN initialization. This saved data (also referred to herein as **checkpoint** data) is used later if the initialization process is resumed after being interrupted (i.e., by a power loss in the storage system)."]

(2) According to Dekoning, two different approaches (programs) are used for host I/O requests. When a host I/O request wants to write data to storage space above the boundary (initialized storage space) or storage space span the boundary, initialization will be performed first and the host I/O request will then be written into the storage space. When a host I/O request wants to write data to storage space fully below the boundary, initialization will not be performed on that storage space, but rather, **it is processed in a special manner that assures integrity of the redundancy data [see abstract]. Specifically, the I/O request is processed by generating new redundancy information consistency with the data to be written by the I/O request (and with any related old, un-initialized data remaining in affected areas**

of the LUN). [See column 9, lines 14-18]

From the above, it can be concluded that aforesaid two manners of Dekoning requires **two different software programs** to perform the I/O requests, disadvantage of which is that two different software programs would stands and occupies system

5   resources of Dekoning and could have negative impact on system efficiency.

(3) Because Dekoning only uses the boundary or checkpoint data to record the degree of initialization, it can not perform the initialization by way of initializing the entire LUN to zero.

10   The problem with initializing the entire LUN to zero is that if data due to a host I/O request have already been stored in the portions of the LUN are fully below the boundary, then when the boundary moves to and even crosses over that portions of LUN in which that data have already been stored, that data will be "initialized to zero", i.e., data loss will be incurred, because without an "initialization progress table that

15   records which portions of LUN have been initialized", the storage system of Dekoning **cannot distinguish portions of a LUN that have been written due to a host I/O request below the boundary from those portions yet to be initialized, but performs "initialization" in the LUN in top to down order.** Dekoning never mentions how to solve this problem. Therefore, the only way to perform initialization

20   in Dekoning is to initialize the LUN by **generating new redundancy information consistency with the all the old data.**

In contrast, according to the present invention, the present invention relates to a method for redundant array of independent disks (RAID) consistency initialization.

25   The method allows fast RAID creation of the RAID subsystem and the RAID is capable of accepting I/O access after completion of the RAID creation. One embodiment of the method comprises the steps of: setting a RAID configuration, creating an initialization progress table (IPT) for storing the progress states of the initialization of the RAID, and writing the IPT and the RAID configuration into a

30   non-volatile memory device. After that, the RAID can accept I/O access and meantime

a consistency initialization can be performed according to the IPT. In one example, the consistency initialization comprises a consecutive initialization mechanism consecutively progressing in background mode and an induced consistency initialization mechanism induced by an I/O accessing to the RAID. In view of the

5   present invention, the key points of the present invention can be summarized as follows:

(1) "initialization progress table (IPT)"- **unlike use of boundary concept of Dekoning to tell initialized portions from un-initialized portions,** the IPT of the

10   present invention **comprises a plurality of N fields to record the initialization state of each initialization region (520)** and is used to distinguish initialized initialization regions from un-initialized initialization regions, in which the initialization progress table includes a plurality of fields, each of which is used to record whether a regional initialization is performed on an initialization region that is associated with one of the

15   fields (please refer to Fig. 5A and paragraph [0048] of the present invention). According to paragraph [0055] of the present invention, "**[0055] Because the consecutive consistency initialization (shown in FIG. 6A), which is performed according to a predetermined schedule**", which means that the consecutive consistency initialization of the present invention is not the same as Dekoning's

20   performing "initialization" from top to down, but performs the initialization in the predetermined schedule. i.e., according to priority and requirements, which means that the present invention does not need to perform the initialization from top to down or from down to top, and thus does not have "boundary concept", but performs the initialization according to the predetermined schedule. To know whether a

25   initialization region has been completed with regional initialization or not, we only have to check the initialization progress table. When the whole system of the present invention is on line, and an I/O from the host makes access to a specific portion the LUN, in contrast to Dekoning, the present invention only requires one software program as below. That is, the present invention only needs only one **Initialization**

30   **software program that performs both "consecutive consistency initialization"**

and "induced consistency initialization" according to the IPT

(a) "consecutive consistency initialization"- before "consecutive consistency initialization" initializes the LUN in a pre-determined schedule, the IPT is looked up to determines whether or not some portions of the LUN have already been initialized.

5   If the IPT shows that those some portions of the LNU have already been initialized, then "consecutive consistency initialization" skips those some portions that have already been initialized, and;

(b) "induced consistency initialization"- when an I/O from the host makes access to a portion of the LUN, which portion is un-initialized and is associated with the I/O,

10   then the induced consistency initialization is induced and is performed in that portion which is associated with the I/O. After that portion that is associated with the I/O is first initialized, then the I/O is executed.

(2) In view of the point (1), it can be concluded as follows.

15   (a) First, for any region, initialization should be performed before a host I/O request is written.

For the above scenario (1)(a), since a consecutive consistency initialization is scheduled, regional initialization will be performed on every scheduled initialization region, and thus before a host I/O is later to make access to the initialization regions

20   that have been performed a consecutive consistency initialization, the associated regions have already been initialized.

For the above scenario (1)(b), because when an I/O from the host makes access to an un-initialized region, a regional initialization will be induced on that region first, and the I/O will be performed later after the induced regional initialization is

25   completed, initialization is always performed before a host I/O request is written.

(b) the present invention only requires **only one software program** to perform initialization before a host I/O request is written in above scenarios (a) and (b), and thus will stand or occupies, in contrast to Dekoning, less system resources.

30   (3) Since the regional initialization is performed according to the initialization

progress table (IPT)", the initialization may be performed either by initializing all regions to zero, or by other manner. In addition, when initialization is performed by initializing all regions to zero, an advantage can be gained that no parity calculation should be performed. On the contrary, according to Dekoning, parity calculation has

5   to be performed when performing the initialization, which will make the performance worse.

Conclusion

From above comparisons and explanations, it is concluded and summarized by the

10   applicant that in contrast to Dekoning, the reasons why the advantages and characteristics of claims of the present invention has non-obviousness are as follows:

(1) initialization progress table (IPT) in amended claim 1 of the present invention- the <u>IPT in amended claim 1 of the present invention is used to tell un-initialized portions</u>

15   <u>of the LUN from initialized portions of the LUN, where the initialization progress table includes a plurality of fields, each of which is used to record whether a regional initialization is performed on an initialization region.</u> In contrast, column 8, lines 30-37 of Dekoning only discloses "the method of Fig. 3 saves checkpoint data as it progresses in the initialization process. **This checkpoint data is used in element 316**

20   **to restore the previous state of the initialization process.** In this manner, a previously interrupted initialization process may be resumed following such an interruption. i.e., power loss", <u>but fails to teach or suggest such an IPT of the present invention, which IPT is used to tell un-initialized portions from initialized portions.</u>

25   (2) For those regions that are to be accessed by the host I/O, the claimed invention will perform regional initialization (induced consistency initialization) on them first, and then perform the host I/O later after the completion of the induced consistency initialization. On the contrary, Dekoning does not perform such an induced initialization on un-initialized regions, but rather performs a special I/O treatment

30   including redundancy data calculation – in Dekoning,"**Where part of an I/O request**

is below the boundary, it is processed in a special manner that assures integrity of the redundancy data."

(3) less software programs – because the initialization progress table (IPT) is used in the claimed invention, only one software program is used to handle host I/O reqeusts. Dekoning, one the contrary, uses two different software programs to handle host I/O requests.

(4) the initialization progress table (IPT) is not obvious to try for persons of ordinary skills in the art.

(a) First, according to Dekoning, no the initialization progress table (IPT) is used, but rather, a checkpoint data is recorded or the boundary concept is utilized. This is what the persons of ordinary skills in the art deems obvious to try. That is, Dekoning invention itself is the best evidence that the use of initialization progress table (IPT) used in the claimed invention is not obvious to try for persons of ordinary skills in the art.

(b) Because of the use of the initialization progress table (IPT), the consecutive initialization of the claimed invention can be scheduled as needed. Dekoning's invention can never perform initialization in a scheduled way.

(c) Because of the use of the initialization progress table (IPT), the design of the claimed invention is different from that of Dekoning, and has the advantages of using only one program and saving the system resources.

(d) According to (b) and (c) the claimed invention generates technical effects that cannot be expected from Dekoning and other prior arts.

Regarding amended independent claim 1

As per amended independent claim 1, it is asserted by the applicant that the amended independent claim 1is non-obvious because of the following reasons:

The amended independent claim 1 claims "creating RAID, including setting a RAID

configuration of the RAID and creating an initialization progress table for storing progress states of the initialization of the RAID; wherein the initialization progress table includes a plurality of fields, each of which is used to record whether a regional initialization is performed on an initialization region"

5   In view of the above-mentioned conclusions, since the amended independent claim 1 is patentably distinct from Dekoning, has overcome the 103 rejection by merging the limitations in the dependent claims, and thus is in condition for allowance, the applicant respectfully requests that rejection be withdrawn.

10   **Regarding amended claim 3**

Amended claim 3 of the present invention claims "wherein the consistency initialization comprises steps of: detecting, when the RAID receives an I/O, whether the initialization region(s) that is(are) associated with the I/O is(are) completed with the regional initialization; and initializing the initialization region(s) first if the

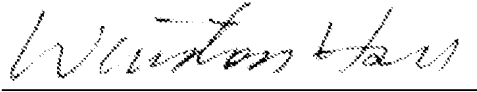15   initialization region(s) is(are) not completed with the regional initialization."

In view of the above-mentioned conclusions, since the amended dependent claim 3 is patentably distinct from Dekoning and is dependent on the independent claim 1 which has overcome the 103 rejection by merging the limitations in the dependent claims, and thus is in condition for allowance, the applicant respectfully requests that rejection

20   be withdrawn, and that the amended dependent claim 3 is in condition for allowance.

Conclusion

Thus, all pending claims are submitted to be in condition for allowance with respect to the cited art for at least the reasons presented above. The Examiner is

25   encouraged to telephone the undersigned if there are informalities that can be resolved in a phone conversation, or if the Examiner has any ideas or suggestions for further advancing the prosecution of this case.

Sincerely yours,

Winston Hsu   Date: 09/25/2008

Winston Hsu, Patent Agent No. 41,526

5   P.O. BOX 506, Merrifield, VA 22116, U.S.A.

Voice Mail: 302-729-1562

Facsimile: 806-498-6673

e-mail : winstonhsu@naipo.com

10   Note: Please leave a message in my voice mail if you need to talk to me. (The time in

D.C. is 12 hours behind the Taiwan time, i.e. 9 AM in D.C. = 9 PM in Taiwan.)